

# Temporal Tessellation for Video Annotation and Summarization

Dotan Kaufman<sup>1</sup>, Gil Levi<sup>1</sup>, Tal Hassner<sup>2,3</sup>, and Lior Wolf<sup>1</sup>

<sup>1</sup>The Blavatnik School of Computer Science, Tel Aviv University, Israel

<sup>2</sup>Information Sciences Institute, USC, CA, USA

<sup>3</sup>The Open University of Israel, Israel

## Abstract

We present a general approach to video understanding, inspired by semantic transfer techniques successfully used for 2D image understanding. Our method considers a video to be a 1D sequence of clips, each one associated with its own semantics. The nature of these semantics – natural language captions or other labels – depends on the task at hand. A test video is processed by forming correspondences between its clips and the clips of reference videos with known semantics, following which, reference semantics can be transferred to the test video. We describe two matching methods, both designed to ensure that (a) reference clips appear similar to test clips and (b), taken together, the semantics of selected reference clips is consistent and maintains temporal coherence. We use our method for video captioning on the LSMDC’16 benchmark and video summarization on the SumMe benchmark. In both cases, our method not only surpasses state of the art results, but importantly, it is the only method we know of that was successfully applied to both video understanding tasks.

## 1. Introduction

Despite decades of research, video understanding problems are still very much at the frontier of computer vision capabilities. The reasons for this are many, and include the challenges of collecting, labeling and processing video data, which is typically much larger yet less abundant than images. It is also at least partially due to the inherent ambiguity of actions in videos which often defy attempts to attach dichotomic labels to video sequences [12].

Rather than attempting to assign videos with single *action labels* (in the same way that 2D images are assigned object classes in, say, the ImageNet collection [35]) a growing number of efforts focus on other representations for the semantics of videos. One popular approach assigns videos

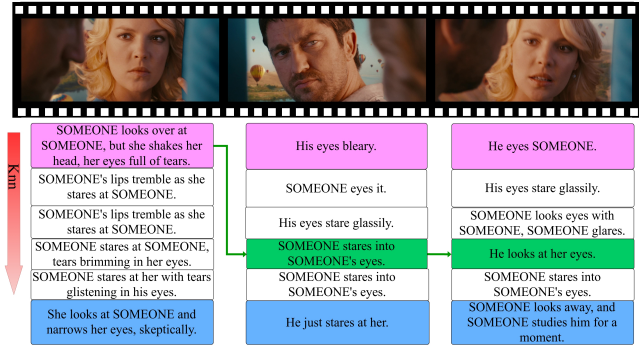


Figure 1. **Tessellation for temporal coherence.** In this example for the task of video captioning, given a query video (top), we seek reference video clips with similar semantics. Tessellation is proposed as a means ensuring that the semantics assigned to the test clip are not only the most relevant (the five options for each clip) but between them best preserve temporal coherence (green path). In blue are the ground truth captions.

with natural language text annotations which describe the events taking place in the video [2, 32]. Systems are then designed to automatically predict these annotations. Others attach video sequences with numeric values indicating what parts of the video are more interesting or important [10]. Machine vision is then expected to determine the relative importance of each part of the video and summarize videos by keeping only their most important parts.

Importantly, although impressive progress was made on these and other video understanding problems, this progress was often made disjointedly: separate specialized systems tailored in order to obtain state of the art performance on different video understanding problems. Still lacking is a *unified* general approach to solving these different tasks.

Our approach is inspired by recent 2D dense correspondence estimation methods (e.g., [14, 25]). These methods were successfully shown to solve a variety of image understanding problems by transferring per-pixel semantics from reference images to query images. This general ap-

proach was effectively applied to a variety of tasks, including single view depth estimation, semantic segmentation and more. We take an analogous approach, applying similar techniques to 1D video sequences rather than 2D images.

Specifically, image based methods combine local, per-pixel appearance similarity with global, spatial smoothness. We instead combine local, per-frame appearance similarity with global semantics smoothness, or *temporal coherence*. This is exemplified in Fig. 1 which shows how temporal coherence improves the text captions assigned to a video sequence.

Our contributions are as follows: **(a)** We describe a novel method for matching test video clips to reference clips. References are assumed to be associated with semantics representing the task at hand; hence, by this matching we transfer semantics from reference to test videos. This process seeks to match clips which share similar appearances while maintaining semantic coherency between the assigned reference clips. **(b)** We discuss two techniques for maintaining temporal coherency: The first uses unsupervised learning for this purpose whereas the second is supervised. **(c)** Finally, we show that our method is general by presenting state of the art results on two recent and challenging video understanding tasks, previously addressed separately: Video caption generation on the LSMDC’16 benchmark [34] and video summarization on the SumMe benchmark [10]. Importantly, we plan to publicly release our code and models.<sup>1</sup>

## 2. Related work

**Dense correspondences for 2D image understanding.** Our work is inspired by methods developed for estimating per-pixel (dense) correspondences between two images of similar, though not necessarily the same, scenes. One of the earliest methods to show potential applications for this capability was [13]. It was later followed by the SIFT flow of [25], which provided a robust means for computing and using dense correspondences.

Briefly, these methods assume that reference images are associated with a *semantic channel* containing, e.g., per pixel depths [13, 18], semantic segmentation labels [24, 42] and even character labels for text images [15]. Robust dense correspondence methods are then applied in order to match the pixels in a novel, test image to these references. Following this, the values of the semantic channel can be transferred to the query. We refer to a recent survey of these methods and their applications in [14]. Surprisingly, despite their potential, we are unaware of previous attempts to apply this approach to videos.

**Video annotation.** Significant progress was made in the relatively short time since work on video annotation / caption generation began. Early methods such as [1, 17, 27, 49]

attempted to cluster captions and videos and applied this for video retrieval. Others [9, 20, 43] generated sentence representations by first identifying semantic video content (e.g., verb, noun, etc.) using classifiers tailored for particular objects and events and then generating template based sentences. Unfortunately, this approach does not scale well, as it requires substantial efforts to provide suitable training data for the classifiers, as well as limits the possible sentences that the model can produce.

More recently, and following the success of image annotation systems based on deep networks such as [6, 48], some began applying similar techniques to videos [6, 40, 46, 50]. Whereas image based methods used convolutional neural networks (CNN) for this purpose, application to video required addressing temporal data, which led to the use of long recurrent neural networks (RNN), particularly short-term memory networks (LSTM) [16]. Our own method uses CNN and LSTM models, but in fundamentally different ways, as we later explain in Sec. 4.

**Video summarization.** This task involves selecting the subset of a query video’s frames which represents its most important content. This is not a new problem and relevant existing work is substantial. Early methods developed for this purpose relied on unsupervised, manually specified cues for determining which parts of a video are important and should be retained. A few such examples include [3, 30, 39, 52].

More recently, the focus shifted towards supervised methods [7, 10, 11, 53], which assume that training videos provide also manually specified labels indicating the importance of different video scenes. These include method which use multiple man-tailored decisions to choose video portions for the summary [10, 11] and methods relying on the determinantal point process (DPP) to increase the diversity of selected video subsets [8, 7, 53].

Contrary to video description tasks, LSTM based methods were only considered for summarization very recently in [54]. Their use of LSTM is also very different from ours.

## 3. Preliminaries

Our approach assumes that test videos are partitioned into clips. It then matches each test clip with a reference (*training*) clip. Matching is performed with two goals in mind: first, at the clip level, we select reference clips with visually similar appearances to the input. Second, at the video level, we select a sequence of clips which best preserves semantic coherency. That is, taken in sequence, the order of selected, reference semantics should adhere to the order in which they appeared in the training videos.

Following this step, the semantics associated with selected reference clips can be transferred to test clips, thereby allowing us to reason about the test video using information

<sup>1</sup>See: [github.com/dot27/temporal-tessellation](https://github.com/dot27/temporal-tessellation)

from our reference. This approach is general, as it allows for different types of semantics to be stored and transferred from reference, training videos to the test videos. This can include, in particular, textual annotations, action labels, manual annotations of interesting frames and more. Thus, different semantics represent different video understanding problems which our method can be used to solve.

### 3.1. Encoding video content

We assume that training and test videos are partitioned into a sequence of clips. A clip  $\mathbf{C}$  consists of a few consecutive frames  $\mathbf{I}_i, i \in 1..n$  where  $n$  is the number of frames in the clip. Our tessellation approach is agnostic to the particular method chosen to represent these clips. Of course, the more robust and discriminative the representation, the better we expect our results to be. We therefore chose the following two step process, based on the recent state of the art video representations of [22].

**Step 1: Representing a single frame.** Given a frame  $\mathbf{I}_i$  we use an off the shelf CNN to encode its appearance. We found the VGG-19 CNN to be well suited for this purpose. This network was recently proposed in [38] and used to obtain state of the art results on the ImageNet, large scale image recognition benchmark (ILSVRC) [35]. In their work, [38] used the last layer of this network to predict ImageNet class labels, represented as one-hot encodings. We instead treat this network as a feature transform function  $f : \mathbf{I} \mapsto \mathbf{a}'$  which for image (frame)  $\mathbf{I}$  returns the 4096D response vector from the penultimate layer of the network.

To provide robustness to local translations, we extract these features by oversampling:  $\mathbf{I}$  is cropped ten times at different offsets around the center of the frame. These cropped frames are normalized by subtracting the mean value of each color channel and then fed to the network. Finally, the ten 4096D response vectors returned by the network are pooled into a single vector by element-wise averaging. Principle component analysis (PCA) is further used to reduce the dimensionality of these features to 500D giving us the final, per frame representation  $\mathbf{a}$ .

**Step 2: Representing multiple frames.** Once frames are encoded, we pool them to obtain a representation for the entire clip. Pooling is performed by Recurrent Neural Network Fisher Vector (RNN-FV) encoding [22].

Specifically, we use their RNN, trained to predict the feature encoding of a future frame,  $\mathbf{a}_i$ , given the encodings for its  $k$  preceding frames,  $(\mathbf{a}_{i-k}, \dots, \mathbf{a}_{i-1})$ . This RNN was trained on the training set from the Large Scale Movie Description Challenge [34], containing roughly 100K videos. We apply the RNN-FV to the representations produced for all the frames in the clip. The gradient of the last layer of this RNN is then taken as a 100,500D representation for the entire sequence of frames in  $\mathbf{C}$ . We again use PCA for dimensionality reduction, this time mapping the features pro-

duced by the RNN-FV to 2,000D dimensions, resulting in our pooled representation  $\mathbf{A}$ . We refer to [22] for more details about this process.

### 3.2. Encoding semantics

As previously mentioned, the nature of the semantics associated with a video depends on the task at hand. We seek a single, effective means of representing these semantics, regardless of the particular task our method is applied to solve. We tested several representations for video semantics and chose the recent Fisher Vector of a Hybrid Gaussian-Laplacian Mixture Model (FV-HGLMM) [19], as it provided the best results in our tests.

Briefly, we assume a textual semantic representation,  $\mathbf{s}$  for a clip  $\mathbf{C}$ , where  $\mathbf{s}$  is a string containing natural language words. We use word2vec [26] to map the sequence of words in  $\mathbf{s}$  to a vector of numbers,  $(s_1, \dots, s_m)$ , where  $m$  is the number of words in  $\mathbf{s}$  and can be different for different clips. FV-HGLMM then maps this sequence of numbers to a vector  $\mathbf{S} \in \mathbb{R}^M$  of fixed dimensionality,  $M$ .

FV-HGLMM is based on the well-known Fisher Vectors (FV) [29, 37, 41]. The standard Gaussian Mixture Models (GMM) typically used to produce FV representations are replaced here with a Hybrid Gaussian-Laplacian Mixture Model which was shown in [19] to be effective for image annotation. More details on this representation can be found in that paper.

### 3.3. The joint semantics video space (SVS)

Clip representations and their associated semantics are all mapped to the joint SVS. We aim to map the appearance of each clip and its assigned semantics to two neighboring points in the SVS. By doing so, given an appearance representation for a query clip, we can search for potential semantic assignments for this clip in our reference set using standard Euclidean distance. This property will later become important in Sec. 4.1.

In practice, all clip appearance representations  $\mathbf{A}$  and their associated semantic representations  $\mathbf{S}$  are jointly mapped to the SVS using regularized Canonical Correlation Analysis (CCA) [47] where the CCA mapping is trained using the ground truth semantics provided with each benchmark. This produces representations  $\mathbf{V}^A$  and  $\mathbf{V}^S$  for the appearance and semantics, respectively, for each clip.

## 4. Tessellation

We assume a data set of training (reference) clips,  $\mathbf{V}_j^A$ , and their associated semantics,  $\mathbf{V}_j^S$ , represented as described in Sec. 3. Here,  $j \in 1..N$  indexes the entire data set containing  $N$  clips. As these clips may come from different videos,  $j$  does not necessarily reflect temporal order.

Given a test video, we process its clips following 3.1 and 3.3, obtaining a sequence of clip representations,  $\mathbf{U}_i^A$  in

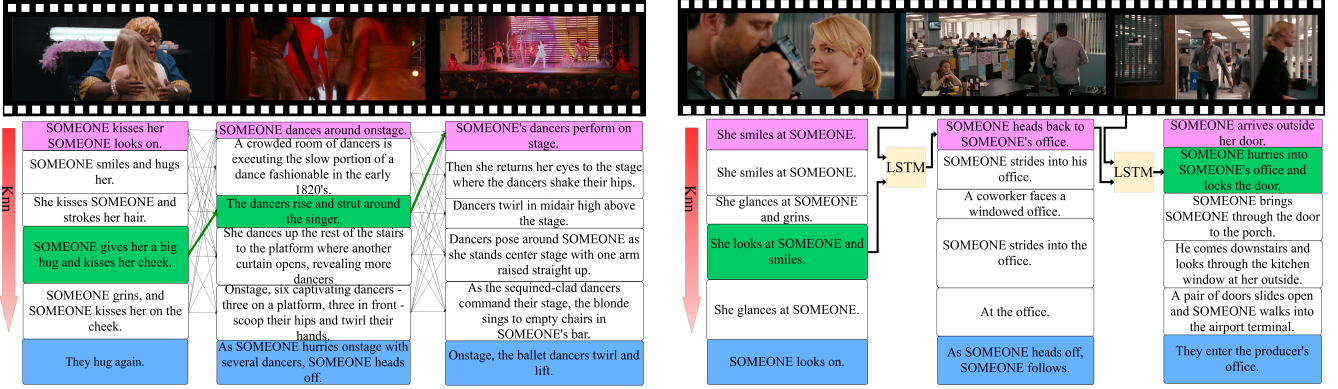


Figure 2. **Our two proposed tessellations.** **Left:** Tessellation by restricted Viterbi. For a query video (top), our method finds visually similar videos and selects the clips that preserve temporal coherence using the Viterbi Method. The ground truth captions are shown in blue, the closest caption is shown in pink. Note that our method does not always select clips with the closest captions but the ones that best preserve temporal coherence. **Right:** Tessellation by predicting the dynamics of semantics. Given a query video (top) and a previous clip selection, we use an LSTM to predict the most accurate semantics for the next clip.

the SVS where consecutive index values for  $i \in M$ , represent consecutive clips in a test video with  $M$  clips. Our goal is to match each  $\mathbf{U}_i^A$  with a data set *semantic* representation  $\mathbf{V}_{j_i}^S$  while optimizing the following two requirements:

1. **Semantics-appearance similarity.** The representation for the test clip *appearance* is similar to the representation of the selected *semantics*.
2. **Temporal coherence.** The selected semantics are ordered similarly to their occurrences in the training set.

Drawing on the analogy to spatial correspondence estimation methods such as SIFT flow [25], the first requirement is our *data term* and the second is a *smoothness term*, albeit with two important distinctions: First, the data term matches test *appearances* to reference *semantics* directly, by building on the joint embedding of semantics and appearances in the SVS. Second, we define the smoothness term in terms of associated semantics and not pixel coordinates.

#### 4.1. Tessellation distribution

Given the sequence of appearance representations  $\mathcal{U} = (\mathbf{U}_1^A, \dots, \mathbf{U}_M^A)$  for the test clip, we seek a corresponding set of reference semantics  $\mathcal{V} = (\mathbf{V}_{j_1}^S, \dots, \mathbf{V}_{j_M}^S)$  (here, again,  $j$  indexes the  $N$  clips in the reference set). We make the Markovian assumption that the semantics assigned to input clip  $i$ , only depends on the appearance of clip  $i$  and the semantics assigned to its preceding clip,  $i - 1$ . This gives the standard factorization of the joint distribution for the clip appearances and their selected semantics:

$$P(\mathcal{V}, \mathcal{U}) = P(\mathbf{V}_{j_1}^S) P(\mathbf{U}_1^A | \mathbf{V}_{j_1}^S) \times \prod_{i=2}^M P(\mathbf{V}_{j_i}^S | \mathbf{V}_{j_{i-1}}^S) P(\mathbf{U}_i^A | \mathbf{V}_{j_i}^S). \quad (1)$$

We set the priors  $P(\mathbf{V}_{j_1}^S)$  to be the uniform distribution. Owing to our mapping of both appearances and semantics to the joint SVS, we can define both posterior probabilities simply using the L2-norm of these representations:

$$P(\mathbf{U}_i^A | \mathbf{V}_{j_i}^S) \propto \exp(-\|\mathbf{U}_i^A - \mathbf{V}_{j_i}^S\|^2) \quad (2)$$

$$P(\mathbf{V}_{j_i}^S | \mathbf{V}_{j_{i-1}}^S) \propto \exp(-\|\mathbf{V}_{j_i}^S - \mathbf{V}_{j_{i-1}}^S\|^2) \quad (3)$$

Ostensibly, we can now apply the standard Viterbi method [31] to obtain a sequence  $\mathcal{V}$  which maximizes this probability. In practice, we used a slightly modified version of this method, and, when possible, a novel method designed to better exploit our training data to predict database matches. These are explained next.

#### 4.2. Restricted Viterbi method.

Given the test clip appearance representations  $\mathcal{U}$ , the Viterbi method provides an assignment  $\mathcal{V}^*$  such that,

$$\mathcal{V}^* = \arg \max_{\mathcal{V}} P(\mathcal{V}, \mathcal{U}). \quad (4)$$

We found that in practice  $P(\mathbf{U}_i^A | \mathbf{V}_{j_i}^S)$  is a long-tail distribution, with only a few database elements  $\mathbf{V}_{j_i}^S$  near enough to any  $\mathbf{U}_i^A$  for their probability to be more than near-zero. We therefore restrict the Viterbi method in two ways. First, we consider only the  $r' = 5$  nearest neighboring database semantics features. Second, we apply a threshold on the probability of our data term, Eq. (2), and do not consider semantics  $\mathbf{V}_{j_i}^S$  falling below this threshold, except for the first nearest neighbor. Therefore, the number of available assignments for each clip is  $1 \leq r \leq 5$ . This process is illustrated in figure 2 (left).



Method	CIDEr-D	BLEU-4	BLEU-1	BLEU-2	BLEU-3	METEOR	ROUGE
BUPT CIST AI lab*	.072	.005	.151	.047	.013	<b>.075</b>	.152
IIT Kanpur*	.042	.004	.116	.003	.011	.070	.138
Aalto University*	.037	.002	.007	.001	.005	.033	.069
Shetty and Laaksonen [36]	.044	.003	.119	.024	.007	.046	.108
Yu et al [51]	.082	.007	.157	.049	<b>.017</b>	.070	.149
S2VT [46]	.088	.007	<b>.162</b>	<b>.051</b>	<b>.017</b>	.070	<b>.157</b>
Proposed Baseline	.098	.007	.144	.042	.016	.056	.130
Unsupervised Tessellation	.102	.007	.146	.043	.016	.055	.137
Supervised Tessellation	<b>.109</b>	<b>.008</b>	.151	.044	<b>.017</b>	.057	.135

Table 1. Video annotation results on the LSMDC’16 challenge [34]. CIDEr-D and BLEU-4 values were found to be the most correlated with human annotations in [33, 45]. Our results on these metrics far outperform others. \* Denotes results which appear in the online challenge result board, but were never published. They are included here as reference.

### 4.3. Predicting the dynamics of semantics

The Viterbi method of Sec. 4.2 is efficient and requires only unsupervised training. Its use of the smoothness term of Eq. (2), however, results in potentially constant semantic assignments, where for any  $j_i$ ,  $\mathbf{V}_{j_i}^S$  can equal  $\mathbf{V}_{j_{i-1}}^S$ .

In cases where reference clips are abundant and come from continuous video sources, we provide a more effective method of ensuring smoothness. This is done by supervised learning of how the semantic labels associated with video clips change over time, and by using that to predict the assignment  $\mathbf{V}_{j_i}^S$  for  $\mathbf{U}_i^A$ .

Our process is illustrated in Fig. 2 (right). We train an LSTM RNN [16] on the semantic and appearance representations of the training set video clips. We use this network as a function:

$$g(\mathbf{V}_0^S, \mathbf{V}_1^S, \dots, \mathbf{V}_{i-1}^S, \mathbf{U}_1^A, \dots, \mathbf{U}_{i-1}^A, \mathbf{U}_i^A) = \mathbf{V}_i^S, \quad \mathbf{V}_0^S = \mathbf{0}, \quad (5)$$

which predicts the semantic representation  $\mathbf{V}_i^S$  for the clip at time  $i$  given the semantic representation,  $\mathbf{V}_{i-1}^S$ , assigned to the preceding clip and the appearance of the current clip,  $\mathbf{U}_i^A$ . The labeled examples used to train  $g$  are taken from the training set, following the processing described in Sec. 3.2 and 3.3 in order to produce 2000D post-CCA vectors. Each pair of previous semantics ground truth and current clip in the training data provides one sample for the training of the LSTM. We employ two hidden layers, each with 2000 LSTM cells. The output, which predicts the semantics of the next clip, is 2000D.

Given a test video, we begin processing it as in Sec. 4.2. In particular, for each of its clip representations  $\mathbf{U}_i^A$ , we select  $r \leq 5$  nearest neighboring semantics from the training set. At each time step  $i$ , we feed the clip and its assigned semantics from the preceding clip at time  $i - 1$  to our LSTM predictor  $g$ . We thus obtain an estimate for the semantics we expect to see at time  $i$ ,  $\hat{\mathbf{V}}_i^S$ .

Of course, the predicted vector  $\hat{\mathbf{V}}_i^S$  cannot necessarily be interpreted as a semantic label, as not all points in the SVS have semantic interpretations (in fact, many instead represent appearance information and more do not represent anything at all). We therefore choose a representation  $\mathbf{V}_{j_i}^S$  out of the  $r$  selected for this clip, such that  $\|\hat{\mathbf{V}}_i^S - \mathbf{V}_{j_i}^S\|^2$  is smallest.

## 5. Results

We next apply the process described above to two separate video understanding tasks: Video Annotation and Video Summarization.

### 5.1. Video Annotation

In our annotation experiments, we used the movie annotation benchmark defined by the 2016 Large Scale Movie Description and Understanding Challenge (LSMDC16) [34]. LSMDC16 presents a unified version of the recently published large-scale movie datasets, M-VAD [44] and MPII-MD [32]. The dataset contains 188 movies, divided to short (4-20 seconds) video clips with associated sentence descriptions. A total of 188 movies are available, from which 152, 12 and 17 are used for training, validation and test, respectively.

Table 1 presents annotation results. We focus primarily on the CIDEr-D [45] and the BLEU-4 [28] measures, as they are the only ones known to be well correlated with human perception [33, 45]. Other metrics are provided because they were previously reported by others. These measures are: BLEU1–3 [28], METEOR [5], and ROUGE-L [23]. We compare our method with several published and unpublished systems. The results include the following three variants of our pipeline.

**Baseline.** Our baseline system using simple nearest neighbor matching to choose reference semantics. We match each test clip with its closest semantics in the SVS space. From Table 1, we see that this simple method alone outper-



Figure 3. **Qualitative video captioning results.** Three caption assignments from our tests on the LSMDC16 benchmark. Each row presents a single query clip. On the right, in green are the captions predicted by our supervised tessellation method; ground truth captions in blue.

forms previous State-of-the-Art. This gain demonstrates the importance of a semantics-appearance similarity matching.

**Unsupervised tessellation.** We tested several configurations of the method described in Sec. 4.2. Specifically we explore different numbers  $r'$  of nearest neighboring database semantics. We found that the best results are achieved with five neighboring database semantics. Evidently (Table 1), even an unsupervised method for considering temporal coherency improves results over our baseline approach and previous methods.

**Supervised tessellation.** To train the LSTM model described in Sec. 4.3 to predict semantics, we sample mini batches of 20 sequences and train for a maximum of 55 epochs using a learning rate decay mechanism. We set the LSTM number of units to 2000, and use two LSTM layers. All hyperparameters were selected using the validation set. This system achieved the overall best performance on both CIDEr-D and BLEU-4, the metrics known to be most correlated with human perception [33, 45], outperforming previous state of the art with a gain of 23% on CIDEr-D. Some qualitative results are provided in Fig. 3.

## 5.2. Video Summarization

We evaluate our proposed Tessellation framework for video summarization on the SumMe benchmark [10]. The benchmark consists of 25 raw user videos each depicting a certain event. The video frames are hand labeled with an importance score ranging from 0 (redundant) and 1 (vital). The videos are about 1-5 minutes in length and the task is to produce a summary in the form of selected frames which is up-to 15% of the given video’s length. The evaluation met-

ric is the average F-measure of the predicted summary with the ground truth annotations. We follow [11, 54] in evaluating with respect to the different provided user annotations.

Similar to video annotation, our approach is to transfer the semantics (here represented by frame importance values) from the gallery to the tessellated video. Our method operates without incorporating additional computational steps, such as optimizing the selected set using the determinantal point processes [21], commonly used for such applications [8, 7, 53].

Table 2 compares our performance with several recent reports on the same benchmark. We again provide results for all three variants of our system. This time, our baseline and the supervised tessellation methods are both outperformed by previous work. Our unsupervised tessellation (Sec. 4.3), however, outperforms the state of the art on this benchmark by a substantial margin.

We believe the simpler unsupervised method worked better here because the available training examples were much fewer than required for the more powerful but more data hungry RNN-LSTM. Specifically, we used only the labels from the SumMe data set, without leveraging other summarization datasets for this purpose (e.g. [4, 39]).

## 6. Conclusions

We present a general tool for video understanding, designed to transfer different types of semantic information from reference, training videos to novel test videos. Two alternatives are proposed: unsupervised tessellation which uses dynamic programming to apply temporal, semantic coherency, and supervised tessellation which employs LSTM

Method	mean F-measure
Gygli et. al [11]	39.7
Long Short-Term Memory [54]	39.8
Summary Transfer [53]	40.9
Proposed Baseline	34.0
Supervised Tessellation	37.8
UnSupervised Tessellation	<b>41.9</b>

Table 2. Video summarization results on the SumMe benchmark [10] Our proposed results outperform previous methods by a substantial margin.

to predict future semantics. We show these methods, coupled with a recent video representation technique, to provide state of the art results on two very different video analysis domains: video annotation and video summarization.

An obvious next step for this approach would be to apply it to other video understanding tasks, primarily action recognition. Existing action recognition benchmarks [12] provide action labels rather than clip semantics and it remains to be seen how well these can be transferred and used to predict action labels.

## References

- [1] H. Aradhye, G. Toderici, and J. Yagnik. Video2text: Learning to annotate video content. In *Int. Conf. on Data Mining Workshops*, pages 144–151. IEEE, 2009.
- [2] D. L. Chen and W. B. Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proc. Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200. Association for Computational Linguistics, 2011.
- [3] W.-S. Chu, Y. Song, and A. Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3584–3592, 2015.
- [4] S. E. F. De Avila, A. P. B. Lopes, A. da Luz, and A. de Albuquerque Araújo. Vsum: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.
- [5] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *In Proceedings of the Ninth Workshop on Statistical Machine Translation*. Citeseer, 2014.
- [6] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 2625–2634, 2015.
- [7] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *Neural Inform. Process. Syst.*, pages 2069–2077, 2014.
- [8] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Large-margin determinantal point processes. *arXiv preprint arXiv:1411.1537*, 2014.
- [9] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proc. Int. Conf. Comput. Vision*, pages 2712–2719, 2013.
- [10] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *European Conf. Comput. Vision*, pages 505–520. Springer, 2014.
- [11] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3090–3098, 2015.
- [12] T. Hassner. A critical review of action recognition benchmarks. In *Proc. Conf. Comput. Vision Pattern Recognition Workshops*, pages 245–250, 2013.
- [13] T. Hassner and R. Basri. Example based 3D reconstruction from single 2D images. In *Proc. Conf. Comput. Vision Pattern Recognition Workshops*. IEEE, 2006.
- [14] T. Hassner and C. Liu. *Dense Image Correspondences for Computer Vision*. Springer, 2015.
- [15] T. Hassner, L. Wolf, and N. Dershowitz. OCR-free transcript alignment. In *Int. Conf. on Document Analysis and Recognition*, pages 1310–1314. IEEE, 2013.
- [16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [17] H. Huang, Y. Lu, F. Zhang, and S. Sun. A multi-modal clustering method for web videos. In *Int. Conf. on Trustworthy Computing and Services*, pages 163–169. Springer, 2012.
- [18] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *Trans. Pattern Anal. Mach. Intell.*, 36(11):2144–2158, 2014.
- [19] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 4437–4446, 2015.
- [20] N. Krishnamoorthy, G. Malkarnenkar, R. J. Mooney, K. Saenko, and S. Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *AAAI Conf. on Artificial Intelligence*, volume 1, page 2, 2013.
- [21] A. Kulesza and B. Taskar. Determinantal point processes for machine learning. *arXiv preprint arXiv:1207.6083*, 2012.
- [22] G. Lev, G. Sadeh, B. Klein, and L. Wolf. RNN fisher vectors for action recognition and image annotation. *arXiv preprint arXiv:1512.03958*, 2015.
- [23] C.-Y. Lin and F. J. Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proc. Annual Meeting on Association for Computational Linguistics*, page 605. Association for Computational Linguistics, 2004.
- [24] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *Trans. Pattern Anal. Mach. Intell.*, 33(12):2368–2382, 2011.
- [25] C. Liu, J. Yuen, and A. Torralba. SIFT flow: Dense correspondence across scenes and its applications. *Trans. Pattern Anal. Mach. Intell.*, 33(5):978–994, 2011.
- [26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

- [27] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, B. Shaw, A. F. Smeaton, and G. Quénot. TRECVID 2012—an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *Proceedings of TRECVID*, 2012.
- [28] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proc. annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.
- [29] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *European Conf. Comput. Vision*, pages 143–156. Springer, 2010.
- [30] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *European Conf. Comput. Vision*, pages 540–555. Springer, 2014.
- [31] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [32] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A dataset for movie description. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2015.
- [33] A. Rohrbach, A. Torabi, T. Maharaj, M. Rohrbach, C. Pal, A. Courville, and B. Schiele. The large scale movie description and understanding challenge (LSMDC 2016), howpublished = Available: <http://tinyurl.com/zabh4et>, month = September, year = 2016.
- [34] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, P. Chris, L. Hugo, C. Aaron, and B. Schiele. Movie description. *arXiv preprint*, 2016.
- [35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, 2015.
- [36] R. Shetty and J. Laaksonen. Video captioning with recurrent networks based on frame-and video-level features and visual content classification. *arXiv preprint arXiv:1512.02949*, 2015.
- [37] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep fisher networks for large-scale image classification. In *Neural Inform. Process. Syst.*, pages 163–171, 2013.
- [38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [39] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 5179–5187, 2015.
- [40] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *Int. Conf. Mach. Learning*, volume 2, 2015.
- [41] V. Sydorov, M. Sakurada, and C. H. Lampert. Deep fisher kernels—end to end learning of the fisher kernel GMM parameters. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1402–1409, 2014.
- [42] M. Tau and T. Hassner. Dense correspondences across scenes and scales. *Trans. Pattern Anal. Mach. Intell.*, 38(5):875–888, 2016.
- [43] J. Thomason, S. Venugopalan, S. Guadarrama, K. Saenko, and R. J. Mooney. Integrating language and vision to generate natural language descriptions of videos in the wild. In *COLING*, page 9, 2014.
- [44] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint*, 2015.
- [45] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 4566–4575, 2015.
- [46] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 4534–4542, 2015.
- [47] H. D. Vinod. Canonical ridge and econometrics of joint production. *Journal of Econometrics*, 4(2):147–166, May 1976.
- [48] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 3156–3164, 2015.
- [49] S. Wei, Y. Zhao, Z. Zhu, and N. Liu. Multimodal fusion for video search reranking. *Trans. on Knowledge and Data Engineering*, 22(8):1191–1199, 2010.
- [50] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *Proc. Int. Conf. Comput. Vision*, pages 4507–4515, 2015.
- [51] Y. Yu, H. Ko, J. Choi, and G. Kim. Video captioning and retrieval models with semantic attention. *arXiv preprint arXiv:1610.02947*, 2016.
- [52] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4):643–658, 1997.
- [53] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video summarization. In *Proc. Conf. Comput. Vision Pattern Recognition*, 2016.
- [54] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *European Conf. Comput. Vision*, 2016.